An Energy Efficient Two-Phase Clocking Scheme

Brad Bridgeman, Yanqing Zhang
ECE 6332 – Fall 2009
University of Virginia
<bcb4nv, yz4hz>@virginia.edu

ABSTRACT

Timing closure problems are commonly solved through using popular place and route software tools such as Cadence Encounter. Though attractive for the sophisticated level of automation, these tools fail to provide an energy efficient design. This is because the tool places excessive buffers either to solve hold time violations, or in generating the clock tree. In this paper, we describe a method to implement an energy efficient two-phase clocking scheme. By cleverly designing the shape of the second phase, we will eliminate the need to insert buffers on logic paths that would violate hold time constraints. We also look at a way to reduce the need for buffering in the clock tree using the two-phase clock.

1. INTRODUCTION

There has been previous literature on clock distribution techniques[1]. Much of the literature has focused on single-phase clocks. However, we will be using a multi-phase clock to solve hold time violations that arise from clock skew. Clock skew is introduced from imbalance in the clock phase delay along different clock routing paths across a circuit design. The skew between two registers in a logic path may be so great that it causes that path to violate hold time. This is conceptually shown in Figure 1, where both registers are transparent for a certain time due to skew. M denotes the master stages, and S denotes the slave stages.

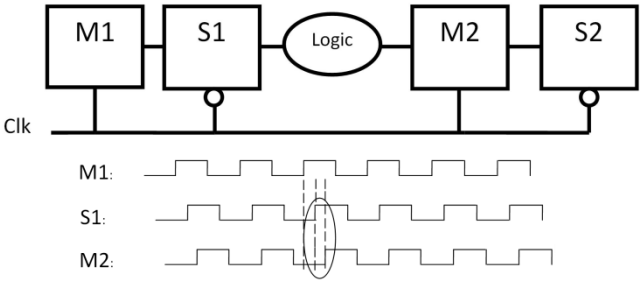

Figure 1. Timing diagram of Clk with skew, hold time problem caused by simultaneous transparent time (circled)

Popular place and route tools are able to select the logic paths that violate hold time constraints, and are able to fix those violations by placing enough buffers between registers. Aside from ensuring proper timing of the circuit, these buffers do not contribute to the functionality while costing area and power. A feasible method to reduce the need for these hold time buffers would be to decrease the skew through de-skewing circuits. However, a quick study of[2] implies that these circuits are power costly. Instead, we use a cleverly designed two-phase clock to eliminate the need for hold time buffers. This clocking scheme is presented in Figure 2.

The pulse shape of the second phase eliminates the overlapping transparent times of the first slave stage and second master stage. Therefore, though skew still exists, it will not cause a malfunctioning logic path.

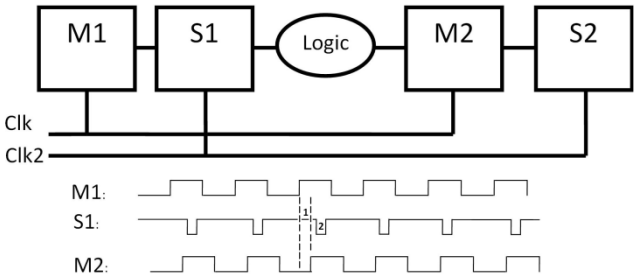

Figure 2. Two-phase clock waveform. Durations of interest are: 1. 1-1 shoulder overlap 2. Negative pulse duration

Place and route tools are also able to conveniently generate clock trees. In clock tree synthesis, buffers are placed to balance the clock phase delay in different clock route paths to minimize the amount of skew. In this way, the skew is limited within a range that can be manually imposed. However, by utilizing the two-phase clocking scheme, the constraint on the maximum skew can be relaxed, thus reducing the need for balancing buffers in the clock tree.

In the following sections, we will describe the circuitry to generate the second phase of the clock, how to implement it in a design, make area and power comparisons between the single and the two-phase clocking scheme, and finally draw conclusions as well as possible topics for future work.

2. IMPLEMENTATION

2.1 Timing Constraint Calculations

We first calculate the timing constraints that we will impose on the circuit. The circuit we use as an experiment for our clocking scheme is a PIC processor formerly done in [3]. This will allow the place and route tool to properly extract the paths that violate hold time and define the waveform of the second phase. Simulation of the registers used in the PIC gives us the setup and hold times for one register. We will use these extracted parameters to calculate our constraints as discussed in the following paragraphs.

The shift register paths are the paths that will easily violate hold time. For a robust hold time fix, we define the necessary hold time as 3x the hold time extracted for one register:

$$t_{hold} = 3t_{hold,1reg} \quad (1)$$

In determining the shape of our second phase clock, we refer back to Figure 2. There are two lengths of time of interest: the 1-1 overlap of the first and second phase clock, which we will call the "shoulder", and the duration of the negative pulse in the second phase. Analysis of the functionality of the clocking scheme shows us that the length of the shoulder should account for the

intolerable amount of skew along the path, which causes hold time violations. For a certain register topology, there is a certain amount of tolerable skew due to the clock-to-q delay. Therefore, the length of the shoulder is given as the difference between the worst case skew and amount of tolerable skew:

$$t_{shoulder} = \delta_{worst} - t_{tolerable} \quad (2)$$

To find the worst case skew, we have the tool synthesize a clock tree. We simulate a discrete register path, increasing the amount of skew on the clock line until the registers do not latch correctly. The results of this are presented in Figure 3.

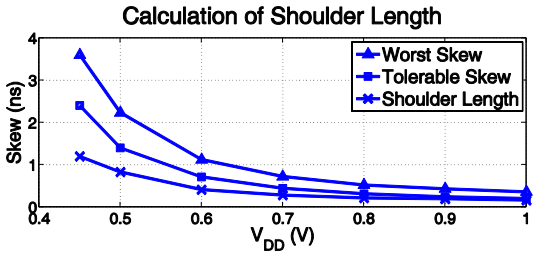

Figure 3. Calculation of Shoulder Length

During the negative pulse, the data is passed from the master stage to the slave stage, and the slave stage latches at the pulse's rising edge. Therefore, it is reasonable to define the length of the pulse as equal to the setup time:

$$t_{pulse} = t_{setup} \quad (3)$$

2.2 Second Phase Clock Generation

We generate the second phase of the clock to meet the criteria described by (2), (3) using a custom pulse generator based on a design presented in [4]. Our design consideration was to limit the power overhead of the pulse generator. To do this we designed the pulse generator using the least gates and minimize sizes as possible. The final design of the pulse generator is presented in Figure 4, with each gate and transistor minimum sized.

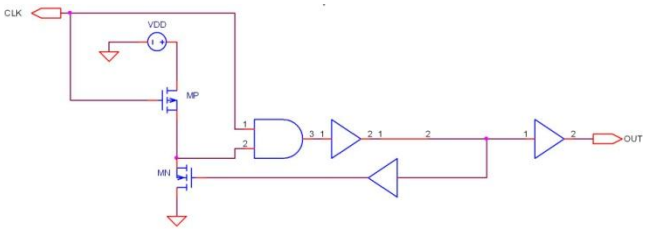

Figure 4. Designed pulse generator

2.3 Critical Hold-Time Path Extraction

We continue to impose the hold time constraint as defined by (1) on the circuit. We have the place and route extract those paths that violate the hold time constraint. Instead of having the tool place buffers along these paths, we adapt the registers along these paths to our two-phase clocking scheme and leave the logic paths as is, which is conceptually shown in Figure 5. Further simulation validates the functionality of these paths which had their hold time violations fixed by our clocking scheme. Therefore, we have eliminated the need for hold time buffers with our clocking scheme.

2.4 Clock Tree Synthesis Considerations

An observation of the clock tree leads to intriguing implications. In generating the clock tree, large buffers are placed to drive the clock signal which may deteriorate across long wire lines, much like as it was mentioned in [5][6]. Smaller buffers placed on subsequent levels of the clock tree are used to balance the phase delay of the clock signal in order to keep the skew within a certain value. We recall that the pulse generator is designed so that the 1-1 shoulder accounts for the worst case skew. Therefore, we can relax the need for the balancing buffers on clock lines that do not have the worst case amount of skew and have the pulse generator account for the skew on that line. We can take out as many of these balancing buffers possible as long as we do not create a skew greater than the original worst case value. Therefore, we have reduced the need for balancing buffers with our clocking scheme. This is conceptually shown in Figure 6.

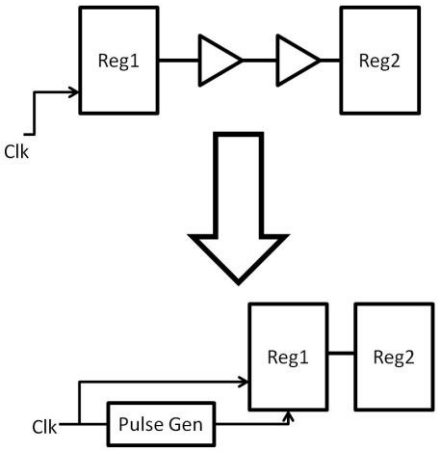

Figure 5. Implemented adaptation of logic paths to two-phase clocking scheme

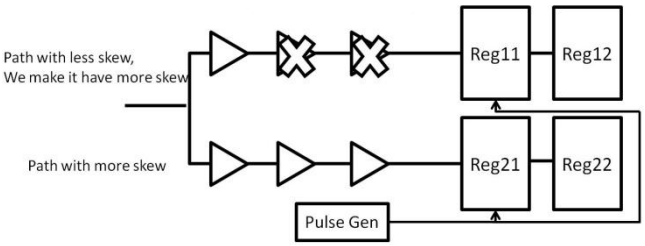

Figure 6. Removing balancing buffers using the two-phase clocking scheme

3. MEASUREMENTS AND COMPARISON

We first evaluate our clocking scheme within a single logic path that violates hold time. We have the place and route tool fix the hold time violation through its buffer insertion method. We compare the power overhead of the buffer insertion method with the two-phase clock method. By comparing each violating path in this way, we are able to measure the number of buffers a pulse generator is equivalent to with respect to power overhead as shown in Figure 7. Therefore, for a *single* logic path, power is saved when the number of buffers inserted to fix hold time violation exceeds the number presented in Figure 7. As an example, the percentage power savings for the worst case path from using the two-phase clocking scheme is presented in Figure 8.

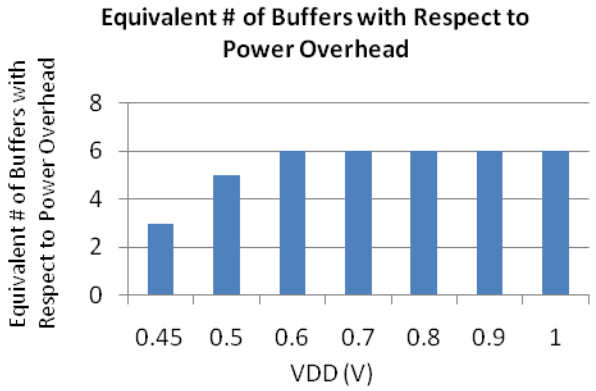

Figure 7. Number of buffers equivalent to power overhead of one pulse generator

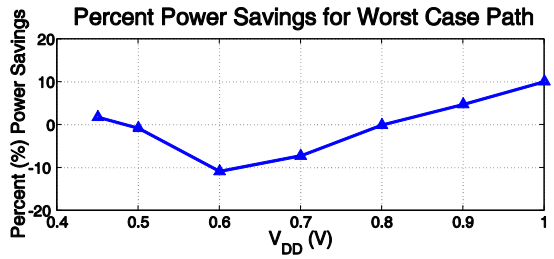

Figure 8. An example of power savings for a single logic path

It is interesting to note that given a certain supply voltage V_{DD}, it may be possible that the two-phase clocking scheme never saves power when considering a single logic path. We observe that no matter if the clocking scheme saves or wastes power, the magnitude of its effectiveness is negligible if we consider only one logic path. We notice that the comparison of area is also negligible in considering a single path because the number of gates used in both methods is approximately the same.

Though it is apparent the two-phase clocking scheme shows promise in saving power, the miniscule effect the clocking scheme has when only a single logic path is considered compels us to explore its effects on multiple logic paths. As mentioned before we have made the analysis that the two-phase clocking scheme can also be utilized to decrease the amount of balancing buffers in the clock tree. For the scope of this paper we will assume that the output of the pulse generator, which is the second phase of the clock, is not buffered to increase its drive ability. Considering the pulse generator is constructed with a buffer, it has a drive ability of its own. We first make the measurement of how many registers, and thus how many paths one pulse generator can drive, the results of which are presented in Figure 9. The percentage savings that are achieved where the pulse generator eliminates the hold buffers in a worst case path and one balancing buffer in each of the paths that it is capable of driving are given in Figure 10. As can be seen, the clocking scheme becomes more attractive in saving power when considered for multiple paths. The amount of area saved is subject to the amount of balance buffers removed and the extra cost of routing the second phase to the respective registers.

4. CONCLUSIONS

In this paper, we have realized a methodology to save power using a cleverly designed two-phase clocking scheme based on a careful analysis of how it can solve hold time violations due to skew without the need of hold time buffers.

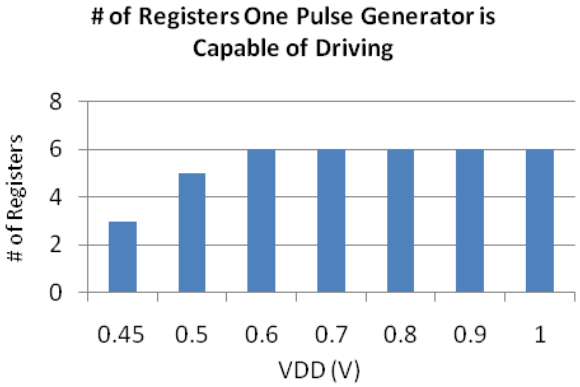

Figure 9. Drive capability of one pulse generator

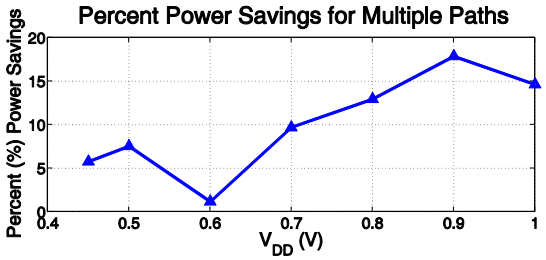

Figure 10. An example of potential power savings when considering multiple paths

We have examined the effects of implementing a two-phase clock with a focus on power savings. Compared to using the traditional method of buffer placement, the two-phase clock demonstrated power savings when appropriately applied. The power savings using this method varied across different supply voltages, and in some cases this demonstrated an increase in the required power. Nevertheless, the clocking scheme is unattractive when applied to single, discrete logic paths. To further increase the power reduction we removed buffers from neighboring paths along the same H-tree branch, and used an existing pulse generator to drive registers on those paths. With this combination, we found that the clocking scheme is more attractive.

5. FUTURE WORK

5.1 Introduce Variation

One concern that was not addressed within the scope of this paper is how to deal with variation. When variation is introduced into the design, the pulse generator will be at a significant risk of failing; this stems primarily from the fact that the current design was not meant to be robust to variation, but simply to provide a minimal operation energy. Even at super-threshold voltages, variation has implications on the functionality of the design that should not be ignored. Monte Carlo simulations, such as the one presented in Figure 11, prove this point. For the two-phase clocking scheme to work, the 1-1 shoulder must be at least 130ps long. However, a quick observation of the Monte Carlo simulation shows that there is a good chance the 1-1 shoulder will not be long enough. When working with variation, a topic of importance is designing for the sub-threshold regime, as will be explained in

the next sub-section. In [7], several concepts are introduced that should be considered when optimizing this design based on variation.

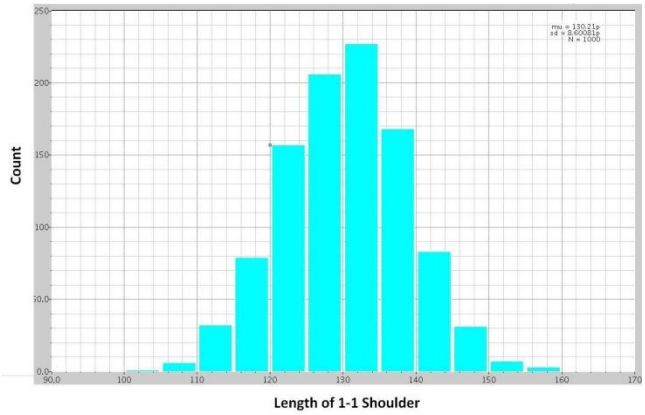

Figure 11. Monte Carlo simulation of 1-1 shoulder length

5.2 Sub-threshold Operation

In the sub-threshold region, a two-phase clocking scheme becomes much more attractive. Variation is more severe in sub-threshold as delay is exponentially dependent upon variation. Since the imbalance in the clock is only a few buffers, the skew may be exponentially greater than expected due to variation. Exacerbating the situation is the delay through several hold time buffers, as their delay may be exponentially less than expected due to variation. Therefore, we safely conclude that it is far more difficult to solve hold time violations using the buffer insertion method in sub-threshold. However, in order to reap the vast savings of the two-phase clocking method, the requirement of a new pulse generator must be introduced.

5.3 Efficient Pulse Generator

In the sub-threshold region, the current pulse generator will begin to fail. This failure is the result of the design being unable to scale to these lower voltages as the increase in delay and variation will consume any resulting waveform. This demonstrates the requirement of a more robust design that can operate under sub-threshold supply voltages and variation.

6. ACKNOWLEDGEMENTS

The authors would like to thank Professor Calhoun, the RLP-VLSI Research Group, and the VLSI 6332/4332 Fall 09 class for their invaluable suggestions and moral support in the completion of this project.

7. REFERENCES

[1] J. K. Das, K. K. Mahapatra, *Low Power Filter Design Using a Novel Dual Edge Triggered Latch*. ICED, 2008.

[2] C. Dike, N. Kurd, P. Patra, J. Barkatullah, *A Design for Digital, Dynamic Clock Deskew*, 2003 Symposium on VLSI Circuits Digest of Technical Papers.

[3] S. Jocke, J. Bolus, S. Wooters, A. Jurik, A. Weaver, T. Blalock, B. Calhoun, *A 2.6-µW Sub-threshold Mixed-signal ECG SoC*, Symposium on VLSI Circuits, 2009.

[4] J. Rabaey, A. Chandrakasan, B. Nikolić. *Digital Integrated Circuits*. 2003.

[5] H. Fair, D. Bailey, *Clocking Design and Analysis for a 600MHz Alpha Microprocessor*, ISSCC98, pp. 398-399, Feb. 1998.

[6] S. Zanella, A. Nardi, A. Neviani, M. Quarantelli, S. Saxena, C. Guardiani, *Analysis of the Impact of Process Variations on Clock Skew*, IEEE Transactions on Semiconductor manufacturing, VOL. 13, NO. 4, November 2000.

[7] J. Kwong, A. Chandrakasan, *Variation-Driven Device Sizing for Minimum Energy Sub-threshold Circuits*, ISLPED06, 2006.